THUẬT TOÁN TRONG AN TOÀN THÔNG TIN

Information Security Algorithms

CHƯƠNG 02

MỘT SỐ THUẬT TOÁN VỀ SỐ NGUYÊN TỐ

Chương 2 – Mục tiêu

- Nắm được các kiến thức cơ bản về số nguyên tố, số giả nguyên tố
- Hiểu và lập trình được một số thuật toán:
  - Sàng nguyên tố
  - Phân tích một số nguyên thành thừa số nguyên tố
  - Kiểm tra một số có là nguyên tố không
  - Sinh số nguyên tố

Bài 01 – Mục tiêu

- Nắm được khái niệm số nguyên tố
- Nắm được kiến thức cơ bản về sàng số nguyên tố Eratosthenes nguyên thủy và sàng phân đoạn
- Nắm được bài toán phân tích một số nguyên ra thừa số nguyên tố
- Hiểu và lập trình được hai thuật toán về sàng Eratosthenes và sàng phân đoạn
- Hiểu và lập trình được thuật toán phân tích một số ra thừa số nguyên tố Pollard’s Rho

Bài 01 – Nội dung

- Kiến thức chung
- Sàng Eratosthenes nguyên thủy
- Sàng Eratosthenes phân đoạn
- Thuật toán Pollard’s Rho

MỘT SỐ THUẬT TOÁN VỀ SỐ NGUYÊN TỐ

- Sàng Eratosthenes
- Phân tích ra thừa số nguyên tố

Kiến thức chung

- Số nguyên tố:
  - Là số tự nhiên có đúng 2 ước số tự nhiên là 1 và chính nó
- Sàng Eratosthenes
  - Là một thuật toán cổ đại để tìm tất cả các số nguyên tố nhỏ hơn hoặc bằng một số nguyên cho trước

Sàng Eratosthenes nguyên thủy

- Ý tưởng:
  - Bắt đầu với số nguyên tố đầu tiên là 2
  - Sinh tất cả các bội của số nguyên tố đã cho (nhỏ hơn hoặc bằng số nguyên cho trước) với hiệu số cố định giữa các số bằng số nguyên tố đó
  - Đánh dấu tất cả các bội của mỗi số nguyên tố là hợp số
  - Các số còn lại đánh dấu là số nguyên tố

Sàng Eratosthenes nguyên thủy

- Bài toán:
  - Tìm tất cả các số nguyên tố nhỏ hơn hoặc bằng số nguyên n cho trước

Sàng Eratosthenes nguyên thủy

- Các bước thực hiện:
  - Bước 1: Liệt kê các số nguyên liên tiếp từ 2 tới n $(2, 3, 4, \dots, n)$
  - Bước 2: Khởi tạo $p = 2$
  - Bước 3: Liệt kê các bội số của p bằng cách đếm các số gia của p từ $2p, 3p, 4p, \dots$ tới n; đánh dấu là hợp số

Sàng Eratosthenes nguyên thủy

- Các bước thực hiện (..)
  - Bước 4: Tìm số nhỏ nhất trong danh sách, lớn hơn p mà không bị đánh dấu
    - Nếu không có số nào thì dừng lại
    - Ngược lại, gán p bằng số vừa tìm được và lặp lại bước 3
  - Bước 5: Kết thúc thuật toán, các số còn lại trong danh sách không bị đánh dấu là tất cả các số nguyên tố nhỏ hơn hoặc bằng n

Sàng Eratosthenes nguyên thủy

- VD: Tìm các số nguyên tố ≤ 30
  - Liệt kê các số nguyên từ 2 tới 30

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

  - $p = 2$, loại bỏ các bội của 2

2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ 9 ~~10~~ 11 ~~12~~ 13 ~~14~~ 15 ~~16~~ 17 ~~18~~ 19 ~~20~~ 21 ~~22~~ 23 ~~24~~ 25 ~~26~~ 27 ~~28~~ 29 ~~30~~

  - $p = 3$, loại bỏ các bội của 3

2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ ~~9~~ ~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~ 19 ~~20~~ ~~21~~ ~~22~~ 23 ~~24~~ 25 ~~26~~ ~~27~~ ~~28~~ 29 ~~30~~

Sàng Eratosthenes nguyên thủy

- VD: Tìm các số nguyên tố ≤ 30 (..)
  - $p = 5$, loại bỏ các bội của 5

2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ ~~9~~ ~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~ 19 ~~20~~ ~~21~~ ~~22~~ 23 ~~24~~ ~~25~~ ~~26~~ ~~27~~ ~~28~~ 29 ~~30~~

  - $p = 7$, loại bỏ các bội của 7, tuy nhiên các bội đó (14, 21, 28) đều đã bị loại
  - ....
  - Tất cả các số còn lại đều là nguyên tố

2 3 5 7 11 13 17 19 23 29

Sàng Eratosthenes nguyên thủy

- Nhận xét:
  - Thuật toán duyệt toàn bộ mảng chứa chuỗi các số không lớn hơn n mà không hiển thị vị trí tham chiếu.
  - Yêu cầu bộ nhớ lớn
    - Với n lớn, bộ nhớ có thể không đáp ứng đủ dãy số nguyên tố
    - Với n vừa phải, việc sử dụng bộ nhớ cache của nó là không tối ưu
  - => Sàng phân đoạn ra đời

Sàng Eratosthenes phân đoạn

- Được biết đến từ những năm 1970
- Mỗi lần chỉ sàng các phần trong phạm vi

Sàng Eratosthenes phân đoạn

- Thuật toán:
  - Bước 1: Chia phạm vi từ 2 tới n thành các đoạn có kích cỡ Δ nào đó, với $\Delta \leq \sqrt{n}$
  - Bước 2: Sử dụng sàng Eratosthenes để tìm các số nguyên tố trong đoạn đầu tiên

Sàng Eratosthenes phân đoạn

- Thuật toán: (..)
  - Bước 3: Theo thứ tự tăng dần, với mỗi đoạn tiếp theo tìm các số nguyên tố như sau, trong đó m là giá trị lớn nhất của đoạn
    - Bước 3.1: Thiết lập một mảng Boolean có kích thước là Δ

Sàng Eratosthenes phân đoạn

- Thuật toán: (..)
  - Bước 3.2: Đánh dấu các vị trí trong mảng không là nguyên tố ứng với các bội của mỗi số nguyên tố $\mathrm{p} \leq \sqrt{m}$ đã tìm được bằng cách tính bội nhỏ nhất của p trong khoảng m- Δ và m, và liệt kê các bội của nó theo các bước của p như bình thường. Các vị trí còn lại tương ứng với các số nguyên tố trong đoạn vì bình phương của một số nguyên tố trong đoạn không thuộc đoạn đó (với $k \geq 1$, ta có $(k\Delta + 1)^2 > (k+1)\Delta$)

MỘT SỐ THUẬT TOÁN VỀ SỐ NGUYÊN TỐ

- Sàng Eratosthenes
- Phân tích ra thừa số nguyên tố

Kiến thức chung

- Bài toán
  - Cho số nguyên dương n, hãy phân tích n ra thừa số nguyên tố, tức là $n = {p_1}^{e_1}.{p_2}^{e_2} \ldots {p_k}^{e_k}$, trong đó p_i là các số nguyên tố đôi một khác nhau và $e_i \geq 1$

Kiến thức chung

- **Ví dụ:** Cho N = 408.508.091, tìm số nguyên tố p, q:

$$p \times q = 408.508.091$$

  - Với máy tính cầm tay $\Rightarrow$ mất bao lâu để có được p, q?
    - Kiểm tra mỗi số nguyên tố xem có là ước của N hay không? Ví dụ: 3, 5, …, cho tới p = 18.313 (số nguyên tố thứ 2000) thì thấy 18.313 thực sự là thừa số của 408.508.091, như vậy dễ dàng xác định được số q = 22.307.
    - Một máy tính kiểm tra 4 số nguyên tố/1 phút $\Rightarrow$ mất 500 phút $\Leftrightarrow$ **hơn 8 giờ** để tìm ra p, q
  - Nếu biết trước giá trị p = 18.313 và q = 22.307 $\Rightarrow$ mất chưa tới **10s** để tính ra N

Kiến thức chung

- ***Lưu ý***:
  - Bài toán quyết định liệu một số nguyên là hợp số hay là nguyên tố nói chung dễ hơn nhiều so với bài toán phân tích ra thừa số
    - => Trước khi phân tích n ra thừa số, nên kiểm tra để đảm bảo n thực sự là hợp số

Kiến thức chung

- ***Định nghĩa 2.2.1***: (Thừa số không tầm thường)

  Phân tích ra thừa số không tầm thường của n là một dạng n=a.b, trong đó 1<a, b<n. Khi đó a, b được gọi là các thừa số không tầm thường.

Kiến thức chung

- ***Nhận xét:***
  - Trong định nghĩa 2.2.1, a,b không nhất thiết phải là số nguyên tố
  - Khi phân tích n ra thừa số nguyên tố, có thể thực hiện:
    - Bước 1: Sử dụng các thuật toán phân tích n thành thừa số
    - Bước 2: Kiểm tra tính nguyên tố của a và b
    - Bước 3: Nếu a hoặc b là hợp số thì tiếp tục áp dụng thuật toán phân tích ra thừa số.

Kiến thức chung

- VD: Áp dụng thuật toán trên phân tích n = 4.337.800 thành tích của các thừa số nguyên tố
  - $4337800 = 2 \times 2168900$ (a = 2, b = 2168900 có là nguyên tố?)
  - $4337800 = 2 \times (2 \times 1084450)$ (kiểm tra 2, 1084450 có là nguyên tố?)
  - $4337800 = 2 \times (2 \times (2 \times 542225))$ (kiểm tra 2, 542225 có là nguyên tố?)
  - $4337800 = 2 \times (2 \times (2 \times (5 \times 108445)))$ (kiểm tra 5, 108445 có là nguyên tố?)
  - ….

Vậy n được phân tích thành tích:

$$4337800 = 2^3 \times 5^2 \times 23^2 \times 41$$

Kiến thức chung

- ***Định nghĩa 2.2.2***: Lũy thừa hoàn hảo
  - Số nguyên n được gọi là một lũy thừa hoàn hảo nếu $n = x^k$ với x, k là các số nguyên thỏa mãn $x \geq 2, k \geq 2$
- ***Chú ý***: Kiểm tra lũy thừa hoàn hảo (perfect power)
  - Nếu $n \geq 2$ có thể kiểm tra xem liệu n có là một lũy thừa hoàn hảo không.
  - Bài toán phân tích n ra thừa số nguyên tố luôn giả thiết n không phải là một lũy thừa hoàn hảo, tức là có ít nhất 2 thừa số nguyên tố khác nhau

Kiến thức chung

- ***Phép chia thử***:
  - Với n là hợp số, trước khi áp dụng các thuật toán phân tích n thành thừa số nguyên tố nên thực hiện phép chia thử n cho tất cả các số nguyên tố "nhỏ"

Thuật toán Pollard's Rho

- Cho $f: S \to S$ là một hàm ngẫu nhiên, trong đó S là một tập hữu hạn của n.
- Cho x_0 là một phần tử ngẫu nhiên của S, xét chuỗi $x_0, x_1, x_2, \ldots$ được xác định bởi $x_{i+1} = \mathrm{f}(x_i)$, với $i \geq 0$
  - Vì S là một tập hữu hạn nên chuỗi phải có chu kì
    - => Tồn tại các chỉ số $i \neq j$, sao cho $x_i = x_j$ (gọi là xảy ra xung đột)

Thuật toán Pollard's Rho

- Vấn đề chuỗi có chu kì liên quan tới một số tấn công thám mã, bao gồm phân tích số nguyên ra thừa số là tìm các chỉ số $i \neq j$, sao cho $x_i = x_j$

Thuật toán Pollard's Rho

- ***Thuật toán Floyd***: (Tìm chu kì)
  - *Bước 1*: Bắt đầu với cặp (x_1, x_2)
  - *Bước 2*: Tính lặp đi lặp lại (x_i, x_{2i}) từ cặp (x_{i-1}, x_{2i-2}) cho tới khi $x_m = x_{2m}$, với giá trị m nào đó

Thuật toán Pollard's Rho

- Cho p là thừa số nguyên tố của hợp số nguyên n. Thuật toán Pollard's Rho phân tích n thành thừa số nguyên tố sẽ tìm sự lặp lại trong chuỗi các số nguyên $x_0, x_1, x_2, \ldots$ được xác định bởi $x_0 = 2$, $x_{i+1} = f(x_i) = {x_i}^2 + 1 \; mod \; p$ với $i \geq 0$
  - Sử dụng thuật toán Floyd tìm x_m và x_{2m} sao cho $x_m \equiv x_{2m} \; (mod \; p)$

Thuật toán Pollard’s Rho

- Vì n chia hết cho p nhưng chưa biết p nên tính $x_i \ mod \ n$ và kiểm tra liệu $\gcd(x_m - x_{2m}, n) > 1$ không. Nếu lại có $\gcd(x_m - x_{2m}, n) < n$ thì thu được thừa số không tầm thường của n.

Thuật toán Pollard’s Rho

- ***Mục tiêu***: Tìm các thừa số nhỏ của một hợp số
- ***Đầu vào***: n là hợp số nhưng n không phải là lũy thừa của một số nguyên tố

Thuật toán Pollard's Rho

- Bước 1: Đặt a=2; b=2
- Bước 2: For i=1,2, … do
  - Bước 2.1: Tính $a = a^2 + 1 \; mod \; n; b = b^2 + 1 \; mod \; n;$ b= $b^2 + 1 \; mod \; n$
  - Bước 2.2: Tính $d = \gcd(a - b, n)$
  - Bước 2.3: If $1 < d < n$ then return(d) và kết thúc với thành công
  - Bước 2.4: If $d = n$ then kết thúc với thất bại

Thuật toán Pollard's Rho

- …
- Thuật toán tìm gcd(a, b):
  - Input: a, b
  - Ouput: gcd(a, b)
    - 1. A=a, B=b
    - 2. while B>0
      - R = A mod B
      - A = B, B = R
    - 3. Return A

Thuật toán Pollard's Rho

- VD: Tính GCD(1970, 1066)?
- 1970 = 1 x 1066 + 904 — gcd(1066, 904)
  1066 = 1 x 904 + 162 — gcd(904, 162)
  904 = 5 x 162 + 94 — gcd(162, 94)
  162 = 1 x 94 + 68 — gcd(94, 68)
  94 = 1 x 68 + 26 — gcd(68, 26)
  68 = 2 x 26 + 16 — gcd(26, 16)
  26 = 1 x 16 + 10 — gcd(16, 10)
  16 = 1 x 10 + 6 — gcd(10, 6)
  10 = 1 x 6 + 4 — gcd(6, 4)
  6 = 1 x 4 + 2 — gcd(4, 2)
  4 = 2 x 2 + 0
  gcd(1970, 1066) = 2

Thuật toán Pollard's Rho

- Ví dụ: Tìm thừa số không tầm thường của $n = 455459$ sử dụng thuật toán Pollard's Rho

a	b	d
5	26	1
26	2871	1
677	179685	1
2871	155260	1
44380	416250	1
179685	43670	1
121634	164403	1
155260	247944	1
44567	68343	743

Vậy ta có:

455459 = 743 × 613

Thuật toán Pollard's Rho

- Chú ý:
  - Thuật toán kết thúc thất bại thì có thể thử lại với hàm khác thay vì $f(x) = x^2 + 1$
  - Chẳng hạn có thể chọn $f(x) = x^2 + c$ với $c \neq 0; -2$

Bài 02 – Mục tiêu

- Hiểu và lập trình được một số thuật toán kiểm tra một số có là nguyên tố hay không
  - Thuật toán kiểm tra Fermat
  - Thuật toán kiểm tra xác suất Miller–Rabin
- Hiểu và lập trình được thuật toán sinh số nguyên tố

Bài 02 – Nội dung

- Kiến thức chung
- Thuật toán kiểm tra Fermat
- Thuật toán kiểm tra xác suất Miller–Rabin
- Thuật toán sinh số nguyên tố

MỘT SỐ THUẬT TOÁN VỀ SỐ NGUYÊN TỐ

- Kiểm tra tính nguyên tố
- Sinh số nguyên tố

Kiến thức chung

- Thuật toán kiểm tra Fermat không thực sự là kiểm tra số nguyên tố dựa trên xác suất vì nó thường thất bại trong việc phân biệt số nguyên tố và một loại hợp số đặc biệt được gọi là số Carmichael

Kiến thức chung

- Các kiểm tra tính nguyên tố theo xác suất có khuôn mẫu sau:
  - Với mỗi số nguyên dương lẻ n, tập $W(n) \subset Z_n$ được xác định thỏa mãn những tính chất sau:
    - Với $a \in Z_n$, có thể kiểm tra xem liệu $a \in W(n)$ không với thời gian đa thức
    - Nếu n là nguyên tố, thì $W(n) = \emptyset$
    - Nếu n là hợp số, thì $\#W(n) \geq \frac{n}{2}$

Kiến thức chung

- ***Định nghĩa 2.3.1***: Nếu n là hợp số thì các phần tử của W(n) được gọi là bằng chứng cho n là hợp số và các phần tử thuộc tập bù $L(n) = Z_n - W(n)$ được gọi là các giá trị đánh lừa (giả - liar)

Kiến thức chung

- Các kiểm tra tính nguyên tố theo xác suất sử dụng tính chất của tập W(n) như sau:
  - Giả sử cần xác định xem n có là một số nguyên tố không?
    - Lựa chọn ngẫu nhiên một số nguyên $a \in Z_n$ và kiểm tra xem liệu $a \in W(n)$ không?
    - => Kết quả n là hợp số nếu $a \in W(n)$ và n là nguyên tố nếu $a \notin W(n)$

Kiến thức chung

- Nếu thực sự $a \in W(n)$ thì n được gọi là **thất bại** với kiểm tra tính nguyên tố đối với cơ sở a
  - => chắc chắn n là hợp số
- Nếu $a \notin W(n)$ thì n được gọi là **qua** với kiểm tra tính nguyên tố đối với cơ sở a
  - => chưa thể khẳng định chắc chắn n là nguyên tố

Kiến thức chung

- ***Định nghĩa 2.3.2***: Một số nguyên n được tin là số nguyên tố theo kiểm tra tính nguyên tố dựa trên xác suất được gọi là số nguyên tố có thể có (có thể là nguyên tố)

Thuật toán kiểm tra Fermat

- ***Định lí Fermat***: Cho n là một số nguyên tố
  - Nếu $\gcd(a, n) = 1$ thì $a^{n-1} \equiv 1 \ (mod\ n)$
- ***Định nghĩa 2.3.3***: Cho n là một hợp số nguyên lẻ. Một số nguyên a, $1 \leq a \leq n - 1$ thỏa mãn $a^{n-1} \not\equiv 1 \ (mod\ n)$ được gọi là bằng chứng Fermat chứng tỏ n là hợp số.

Thuật toán kiểm tra Fermat

- ***Định nghĩa 2.3.4***: Cho n là một hợp số nguyên lẻ và cho a là một số nguyên, $1 \leq a \leq n-1$. Thì n được gọi là một số giả nguyên tố với cơ sở a nếu $a^{n-1} \equiv 1 \ (mod\ n)$. Số nguyên a được gọi là một giá trị đánh lừa cho tính nguyên tố của n.
  - VD: $n = 341\ (= 11 \times 31)$ là một số giả nguyên tố đối với cơ sở $a = 2$ vì $2^{340} \equiv 1 \ (mod\ 341)$

Thuật toán kiểm tra Fermat

- ***Thuật toán FERMAT(n,t)***: Kiểm tra xem liệu n có là số nguyên tố?
  - Đầu vào: n là số nguyên lẻ, $n \geq 3$ và tham số an toàn t ≥ 1
  - Đầu ra: Hoặc "nguyên tố" hoặc "hợp số"

Thuật toán kiểm tra Fermat

- ***Thuật toán FERMAT(n,t)***: Kiểm tra xem liệu n có là số nguyên tố?
  - Bước 1: For i = 1 to t do
    - Bước 1.1 Chọn ngẫu nhiên số nguyên $a, 2 \leq a \leq n-2$
    - Bước 1.2: Sử dụng thuật toán nhân bình phương có lặp tính $r = a^{n-1} \ (mod\ n)$
    - Bước 1.3 Nếu $r \neq 1$ thì return (“Hợp số”)
  - Bước 2: Return (“Nguyên tố”)

Thuật toán kiểm tra Fermat

- ***Nhận xét***:
  - Kết quả của thuật toán FERMAT(n,t) là hợp số thì chắc chắn n là hợp số
  - Ngược lại, không có bằng chứng chứng tỏ n thật sự là nguyên tố

Thuật toán kiểm tra Fermat

- ***Nhận xét***: (..)
  - Tuy nhiên, vì các số giả nguyên tố đối với cơ sở a được biết là rất hiếm nên kiểm tra Fermat cung cấp câu trả lời đúng cho hầu hết các đầu vào
    - Nhưng không có nghĩa là cung cấp câu trả lời đúng cho hầu hết các lần trên mọi đầu vào

Thuật toán kiểm tra Fermat

- Thuật toán nhân bình phương có lặp

Input

$a \in Z_n$ và số nguyên k, $0 \le k < n$ có biểu diễn nhị phân:

$$k = \sum_{i=0}^{t} k_i 2^i$$

Output

$a^k \bmod n$

Thuật toán nhân bình phương có lặp

(1). Đặt b ← 1
Nếu k = 0 thì
Return (b)

(2). Đặt A ← a

(3). Nếu $k_0 = 1$ thì đặt b ← a

(4). For i from 1 to t do
4.1. Đặt A ← A^2 mod n
4.2. Nếu $k_i = 1$ thì b ← A.b mod n

(5). Return (b)

Bài tập áp dụng:

- 41^{101} mod 211
- 5^{596} mod 1234 = ?

Thuật toán nhân bình phương có lặp

- **Giải:**
  - Ta phân tích $101 = 2^6 + 2^5 + 2^2 + 2^0$. Áp dụng phương pháp nhân và bình phương có lặp ta có bảng giá trị sau:

i	0	1	2	3	4	5	6
k_i	1	0	1	0	0	1	1
A	41	204	49	80	70	47	99
b	41	411	110	110	110	106	**155**

Thuật toán kiểm tra Fermat

- BT áp dụng thuật toán Fermat kiểm tra số n = 383 có là số nguyên tố hay ko? (cho t = 2)

Thuật toán kiểm tra Fermat

- **Giải:**
  - t = 1:
    - Chọn a = 2 (thỏa mãn $2 \leq a \leq 383 - 2$)
    - áp dụng tt nhân bình phương có lặp tính r = 2^{382} mod 383
      - Phân tích 382 = $2^8 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1$

i	0	1	2	3	4	5	6	7	8
k_i	0	1	1	1	1	1	1	0	1
A	2	4	16	256	43	317	143	150	286
b	1	4	64	298	175	323	229	229	**1**

Vậy ta có:
r= 2^{382} mod 383 = 1

Thuật toán kiểm tra Fermat

- **Giải:**
  - Tương tự với t = 2:
    - Chọn a = 3 (thỏa mãn $2 \leq a \leq 383 - 2$)
    - áp dụng tính bằng tt nhân bình phương ta có: $r = 3^{382} \bmod 383 = 1$
  - Với 2 vòng lặp ta đều có r = 1. Vậy Return (“Nguyên tố”)

Thuật toán kiểm tra Fermat

- ***Định nghĩa 2.3.5***: Số Carmichael
  - Số Carmichael n là một hợp số nguyên thỏa mãn $a^{n-1} \equiv 1 \ (mod \ n)$

    với tất cả các số nguyên a thỏa mãn $\gcd(a, n) = 1$

Thuật toán kiểm tra xác suất Miller–Rabin

- ***Khẳng định 2.3.2***: Cho x, y và n là các số nguyên. Nếu $x^2 \equiv y^2 \ (mod\ n)$ nhưng $x \not\equiv \pm y \ (mod\ n)$ thì $\gcd(x - y, n)$ là thừa số không tầm thường của n.
  - Ví dụ?
    - Với n = 30, x = 7; y = 13 ta có $7^2 \equiv 13^2 \bmod 30$ ($\equiv 19 \bmod 30$)

      nhưng $7 \not\equiv \pm 13 \ (mod\ n)$, do đó gcd(7-13, 30) = 6 là thừa số không tầm thường của 30.

Thuật toán kiểm tra xác suất Miller–Rabin

- ***Khẳng định 2.3.3***: Cho n là một số nguyên tố lẻ, và cho $n - 1 = 2^s r$, trong đó r là một số lẻ. Cho a là một số nguyên bất kì thỏa mãn $\gcd(a, n) = 1$. Thì với j nào đó, $0 \leq j \leq s - 1$:
  - Hoặc $a^r \equiv 1 \ (mod\ n)$
  - Hoặc $a^{2^j r} \equiv -1 \ (mod\ n)$

Thuật toán kiểm tra xác suất Miller–Rabin

- ***Định nghĩa 2.3.6***: Cho n là một hợp số nguyên lẻ và cho $n-1=2^s r$, trong đó r là lẻ. Cho a là một số nguyên trong đoạn $[1, n-1]$.
  - (i) Nếu $a^r \not\equiv 1 \ (mod\ n)$ và nếu $a^{2^j r} \not\equiv -1 \ (mod\ n)$ với tất cả j, $0 \leq j \leq s-1$ thì a được gọi là bằng chứng mạnh chứng tỏ n là hợp số.

Thuật toán kiểm tra xác suất Miller–Rabin

- ***Định nghĩa 2.3.6***: (..)
  - (ii) Ngược lại, chẳng hạn nếu hoặc $a^r \equiv 1 \ (mod\ n)$ hoặc $a^{2^j r} \equiv -1 \ (mod\ n)$ với j nào đó, $0 \leq j \leq s-1$, thì n được gọi là số giả nguyên tố mạnh đối với cơ sở a (Tức là n hoạt động như một số nguyên tố). Số nguyên a được gọi là giá trị đánh lừa mạnh cho tính nguyên tố của n.

Thuật toán kiểm tra xác suất Miller–Rabin

- VD: Số giả nguyên tố mạnh
  - Xét hợp số nguyên $n = 91\ (= 7 \times 13)$. Vì 91 – 1 = 90 = 2 × 45

=> s = 1 và r = 45.

  - Vì $9^r = 9^{45} \equiv 1\ (mod\ 91)$ nên 91 là một số giả nguyên tố mạnh đối với cơ sở 9
  - Tập tất cả các giá trị đánh lừa mạnh của 91 là:

$\{1, 9, 10, 12, 16, 17, 22, 29, 38, 53, 62, 69, 74, 75, 79, 81, 82, 90\}$

Thuật toán kiểm tra xác suất Miller–Rabin

- ***Thuật toán MILLER-RABIN(n,t)***: Kiểm tra xem liệu n có là số nguyên tố?
  - Đầu vào: Một số nguyên lẻ $n \geq 3$ và tham số an toàn $t \geq 1$
  - Đầu ra: Hoặc "nguyên tố" hoặc "hợp số"

Thuật toán kiểm tra xác suất Miller–Rabin

- ***Thuật toán MILLER-RABIN(n,t)***: Kiểm tra xem liệu n có là số nguyên tố?
  - Bước 1: Viết $n - 1 = 2^s r$ để r là lẻ
  - Bước 2: For i = 1 to t do
    - Bước 2.1: Chọn ngẫu nhiên một số nguyên $a, 2 \leq a \leq n - 2$
    - Bước 2.2: Sử dụng thuật toán nhân bình phương có lặp tính $y = a^r \ mod \ n$

Thuật toán kiểm tra xác suất Miller–Rabin

- ***Thuật toán MILLER-RABIN(n,t)***: Kiểm tra xem liệu n có là số nguyên tố?
  - Bước 2.3: Nếu $y \neq 1$ và $y \neq n-1$ thì
    - $j = 1$
    - While j $\leq s-1$ và $y \neq n-1$ do
      - Tính $y = y^2 \, mod \, n$
      - Nếu $y = 1$ thì return(“hợp số”)
      - $j = j + 1$
    - Nếu $y \neq n-1$ thì return(“hợp số”)
  - Bước 3: Return (“nguyên tố”)

Thuật toán kiểm tra xác suất Miller–Rabin

- ***Nhận xét***:
  - Thuật toán MILLER-RABIN(n,t) kiểm tra xem liệu với mỗi cơ sở a có thỏa mãn định nghĩa 2.3.6 không?
  - Dòng lệnh 5 của bước 2.3 nếu $y = 1$ thì $a^{2^j r} \equiv 1 \ (mod\ n)$. Vì đây cũng là trường hợp mà $a^{2^{j-1} r} \not\equiv \pm 1 \ (mod\ n)$ nên nó tuân theo Khẳng định 2.3.2 mà n là hợp số (thực tế là $\gcd(a^{2^j r} - 1, n)$ là thừa số không tầm thường của n)

Thuật toán kiểm tra xác suất Miller–Rabin

- ***Nhận xét***: (..)
  - Dòng lệnh thứ 7 của bước 2.3, nếu $y \neq n-1$ thì a là bằng chứng mạnh đối với n.
  - Nếu thuật toán MILLER-RABIN(n,t) trả về kết quả là "hợp số" thì n chắc chắn là hợp số vì các số nguyên tố không được vi phạm Khẳng định 2.3.3
  - Nếu n **thực sự** là "nguyên tố" thì thuật toán luôn trả lại kết quả là "nguyên tố"

Thuật toán kiểm tra xác suất Miller–Rabin

- ***Nhận xét***: (..)
  - ...
  - Người ta chứng minh được rằng xác suất để số giả nguyên tố đó không là số nguyên tố là ¼. Suy ra nếu lặp t phép thử với các lựa chọn ngẫu nhiên khác nhau của số a, thì khi đó xác suất để số n sau t phép thử là số nguyên tố là: $1-(1/4)^t$
    - **Ví dụ:** Sau 10 bước, t = 10, mà số đã cho n đều có thể là nguyên tố, thì xác suất để n là số nguyên tố là $1 – (1/4)^{10} > 0.99999$.

Thuật toán kiểm tra xác suất Miller–Rabin

- **VD:** áp dụng thuật toán Miller Rabin kiểm tra n = 383 có là nguyên tố hay không?
  - **Giải:**
    - Ta có n – 1 = 382 = 2^1. 191 (s = 1; r = 191)
    - Chọn 2 lần lặp t (t = 2):
      - i = 1: chọn a = 2 (thỏa mãn $2 \leq a \leq n-2$)
        - Áp dụng tt nhân bình phương có lặp tính: $y = a^r \ mod \ n = 2^{191} mod \ 383 = 1$
        - Thấy y = 1 => không thực hiện các lệnh mục 2.3

Thuật toán kiểm tra xác suất Miller–Rabin

- **Giải:**
  - ...
  - i= 2
    - Chọn a = 5 (thỏa mãn $2 \leq a \leq n-2$)
      - Áp dụng tt nhân bình phương có lặp tính: $y = a^r \ mod \ n = 5^{191} mod \ 383 = 382$
      - Thấy y =382 = n - 1 => không thực hiện các lệnh mục 2.3
  - Kết thúc vòng lặp. Return (“Nguyên tố”)

Thuật toán kiểm tra xác suất Miller–Rabin

- **BTVN:** Áp dụng thuật toán Miller – Rabin kiểm tra n = 57 có là nguyên tố?
  - **Giải:**
    - Ta có n – 1 = 56 = 2^3. 7 (s = 3; r = 7)
    - Chọn 2 lần lặp t (t = 2):
      - i = 1: chọn a = 2 (thỏa mãn $2 \leq a \leq n-2$)
        - Áp dụng tt nhân bình phương có lặp tính: $y = a^r \, mod \, n = 2^7 mod \, 57 = 14$

Thuật toán kiểm tra xác suất Miller–Rabin

- ...
  - $y = 14 \neq 1$ và $\neq 36$; $j = 1$
    - j= 1, y = 14 (thỏa mãn đk vòng lặp $j \leq 2$ và $y \neq 36$)
      - $y = 14^2 \bmod 57 = 25 \neq 1$
      - j = j+1 = 2 và $y \neq 36$ (thỏa mãn đk vòng lặp)
        - $y = 25^2 \bmod 57 = 55 \neq 1$
        - j = j + 1 = 3 (không thỏa mãn đk vòng lặp $j \leq 2$ => dừng vòng lặp while)
      - Ta có $y = 55 \neq (n - 1) = 36 \Rightarrow$ **Return ("Hợp số")**

MỘT SỐ THUẬT TOÁN VỀ SỐ NGUYÊN TỐ

- Kiểm tra tính nguyên tố
- Sinh số nguyên tố

Kiến thức chung

- Bài toán sinh số nguyên tố khác với bài toán kiểm tra tính nguyên tố của một số, nhưng thường liên quan đến bài toán kiểm tra tính nguyên tố

Kiến thức chung

- ***Định lí 2.4.1***: (Định lí số nguyên tố)

  Cho $\pi(x)$ kí hiệu là số các số nguyên tố $\leq x$, thì $\lim_{x\to\infty} \frac{\pi(x)}{x/lnx} = 1$

- ***Nhận xét***:

  - Từ định lí 2.4.1 suy ra tỉ lệ số nguyên dương $\leq x$ là nguyên tố xấp xỉ là $1/lnx$

    - Vì một nửa các số nguyên $\leq x$ là chẵn nên tỉ lệ các số nguyên lẻ $\leq x$ là nguyên tố xấp xỉ là $2/lnx$

Kiến thức chung

- => Chiến lược chọn một số nguyên tố có thể k-bit ngẫu nhiên là chọn lặp đi lặp lại một cách ngẫu nhiên các số nguyên lẻ k-bit n cho đến khi được một số n mà thuật toán MILLER-RABIN(n,t) cho kết quả là nguyên tố với một tham số an toàn t thích hợp.

Kiến thức chung

- Vì xác xuất một số nguyên ngẫu nhiên n có một ước nguyên tố nhỏ là tương đối lớn nên trước khi áp dụng kiểm tra Miller-Rabin nên chia thử n với các số nguyên tố nhỏ hơn một giới hạn B được xác định trước, có thể thực hiện:
  - Chia n cho tất cả các số nguyên tố nhỏ hơn B
  - Hoặc tính ước chung lớn nhất của n và các tích của một vài các số nguyên tố $\leq B$

Thuật toán sinh số nguyên tố

- ***Thuật toán RANDOM-SEARCH(k,t)***: Tìm kiếm ngẫu nhiên một số nguyên tố sử dụng kiểm tra Miller-Rabin
  - ***Đầu vào***: Một số nguyên k, tham số an toàn t
  - ***Đầu ra***: Một số nguyên tố có thể k-bit

Thuật toán sinh số nguyên tố

- ***Thuật toán RANDOM-SEARCH(k,t)***: (..)
  - *Bước 1*: Sinh ngẫu nhiên một số nguyên n k-bit
  - *Bước 2*: Sử dụng phép chia thử để xác định liệu n có chia hết cho một số nguyên tố bất kì $\leq B$ không.
    - Nếu n chia hết thì quay lại Bước 1
  - *Bước 3*: Nếu MILLER-RABIN(n,t) trả về kết quả "nguyên tố" thì return(n). Ngược lại, quay lại Bước 1